

Algorithms for Glushkov \mathbb{K} -graphs

Pascal CARON*, Marianne FLOURET [†]

LITIS, Université de Rouen, 76801 Saint Étienne du Rouvray, France

LITIS, Université du Havre, 76058 Le Havre Cedex, France

October 29, 2018

Abstract

The automata arising from the well known conversion of regular expression to non deterministic automata have rather particular transition graphs. We refer to them as the Glushkov graphs, to honour his nice expression-to-automaton algorithmic short cut [8]. The Glushkov graphs have been characterized [5] in terms of simple graph theoretical properties and certain reduction rules. We show how to carry, under certain restrictions, this characterization over to the weighted Glushkov graphs. With the weights in a semiring \mathbb{K} , they are defined as the transition Glushkov \mathbb{K} -graphs of the Weighted Finite Automata (WFA) obtained by the generalized Glushkov construction [4] from the \mathbb{K} -expressions. It works provided that the semiring \mathbb{K} is factorial and the \mathbb{K} -expressions are in the so called star normal form (SNF) of Brüggeman-Klein [2]. The restriction to the factorial semiring ensures to obtain algorithms. The restriction to the SNF would not be necessary if every \mathbb{K} -expressions were equivalent to some with the same literal length, as it is the case for the boolean semiring \mathbb{B} but remains an open question for a general \mathbb{K} .

Keywords: Formal languages, weighted automata, \mathbb{K} -expressions.

1 Introduction

The extension of boolean algorithms (over languages) to multiplicities (over series) has always been a central point in theoretical research. First, Schützenberger [15] has given an equivalence between rational and recognizable series extending the classical result of Kleene [11]. Recent contributions have been done in this area, an overview of knowledge of these domains is presented by Sakarovitch in [14]. Many research works have focused on producing a small WFA. For example, Caron and Flouret have extended the Glushkov construction to WFAs [4]. Champarnaud *et al* have designed a quadratic algorithm [?] for

*Pascal.Caron@univ-rouen.fr

[†]Marianne.Flouret@univ-lehavre.fr

computing the equation WFA of a \mathbb{K} -expression. This equation WFA has been introduced by Lombardy and Sakarovitch as an extension of Antimirov's algorithm [12] based on partial derivatives.

Moreover, the Glushkov WFA of a \mathbb{K} -expression with n occurrences of symbol (we say that its alphabetic width is equal to n) has only $n + 1$ states; the equation \mathbb{K} -automaton (that is a quotient of the Glushkov automaton) has at most $n + 1$ states.

On the opposite, classical algorithms compute \mathbb{K} -expressions the size of which is exponential with respect to the number of states of the WFA. For example, let us cite the block decomposition algorithm proven in [1].

In this paper, we also address the problem of computing short \mathbb{K} -expressions, and we focus on a specific kind of conversion based on Glushkov automata. Actually the particularity of Glushkov automata is the following: any regular expression of width n can be turned into its Glushkov $(n + 1)$ -state automaton; if a $(n + 1)$ -state automaton is a Glushkov one, then it can be turned into an expression of width n . The latter property is based on the characterization of the family of Glushkov automata in terms of graph properties presented in [5]. These properties are stability, transversality and reducibility. Brüggemann-Klein defines regular expressions in *Star Normal Form* (SNF) [2]. These expressions are characterized by underlying Glushkov automata where each edge is generated exactly one time. This definition is extended to multiplicities. The study of the SNF case would not be necessary if all \mathbb{K} -expressions were equivalent to some in SNF with the same litteral length, as it is the case for the boolean semiring \mathbb{B} .

The aim of this paper is to extend the characterization of Glushkov automata to the multiplicity case in order to compute a \mathbb{K} -expression of width n from a $(n + 1)$ -state WFA. This extension requires to restrict the work to factorial semirings as well as Star Normal Form \mathbb{K} -expressions.

We exhibit a procedure that, given a WFA M on \mathbb{K} a factorial semiring, outputs the following: either M is obtained by the Glushkov algorithm from a proper \mathbb{K} -expression E in Star Normal Form and the procedure computes a \mathbb{K} -expression F equivalent to E , or M is not obtained in that way and the procedure says no.

The following section recalls fundamental notions concerning automata, expressions and Glushkov conversion for both boolean and multiplicity cases. An error in the paper by Caron and Ziadi [5] is pointed out and corrected. The section 3 is devoted to the reduction rules for acyclic \mathbb{K} -graphs. Their efficiency is provided by the confluence of \mathbb{K} -rules. The next section gives orbit properties for Glushkov \mathbb{K} -graphs. The section 5 presents the algorithms computing a \mathbb{K} -expression from a Glushkov \mathbb{K} -graph and details an example.

2 Definitions

2.1 Classical notions

Let Σ be a finite set of letters (alphabet), ε the empty word and \emptyset the empty set. Let $(\mathbb{K}, \oplus, \otimes)$ be a zero-divisor free semiring where $\bar{0}$ is the neutral element of (\mathbb{K}, \oplus) and $\bar{1}$ the one of (\mathbb{K}, \otimes) . The semiring \mathbb{K} is said to be zero-divisor free [9] if $\bar{0} \neq \bar{1}$ and if $\forall x, y \in \mathbb{K}$, $x \otimes y = \bar{0} \Rightarrow x = \bar{0}$ or $y = \bar{0}$.

A *formal series* [1] is a mapping S from Σ^* into \mathbb{K} usually denoted by $S = \sum_{w \in \Sigma^*} S(w)w$

where $S(w) \in \mathbb{K}$ is the coefficient of w in S . The *support* of S is the language $Supp(S) = \{w \in \Sigma^* | S(w) \neq \bar{0}\}$.

In [12], Lombardy and Sakarovitch explain in details the computation of \mathbb{K} -expressions. We have followed their model of grammar. Our constant symbols are ε the empty word and \emptyset . Binary rational operations are still $+$ and \cdot , the unary ones are Kleene closure $*$, positive closure $^+$ and for every $k \in \mathbb{K}$, the multiplication to the left or to the right of an expression \times . For an easier reading, we will write kE (respectively Ek) for $k \times E$ (respectively $E \times k$). Notice that our definition of \mathbb{K} -expressions, which set is denoted $E_{\mathbb{K}}$, introduces the operator of positive closure. This operator preserves rationality with the same conditions (see below) that the Kleene closure's one.

\mathbb{K} -expressions are then given by the following grammar:

$$E \rightarrow a \in \Sigma \mid \emptyset \mid \varepsilon \mid (E + E) \mid (E \cdot E) \mid (E^*) \mid (E^+) \mid (kE), k \in \mathbb{K} \mid (Ek), k \in K$$

Notice that parenthesis will be omitted when not necessary. The expressions E^+ and E^* are called *closure expressions*. If a series S is represented by a \mathbb{K} -expression E , then we denote by $c(S)$ (or $c(E)$) the coefficient of the empty word of S . A \mathbb{K} -expression E is

valid [14] if for each closure subexpression F^* and F^+ of E , $\sum_{i=0}^{+\infty} c(F^i) \in \mathbb{K}$.

A \mathbb{K} -expression E is *proper* if for each closure subexpression F^* and F^+ of E , $c(F) = \bar{0}$.

We denote by $\mathcal{E}_{\mathbb{K}}$ the set of proper \mathbb{K} -expressions. Rational series can then be defined as formal series expressed by *proper* \mathbb{K} -expressions. For E in $\mathcal{E}_{\mathbb{K}}$, $Supp(E)$ is the support of the rational series defined by E .

The length of a \mathbb{K} -expression E , denoted by $||E||$, is the number of occurrences of letters and of ϵ appearing in E . By opposition, the *litteral length*, denoted by $|E|$ is the number of occurrences of letters in E . For example, the expression $E = (a + 3)(b + 2) + (-1)$ as a length of 5 and a litteral length of 2.

A *weighted finite automaton* (WFA) on a zero-divisor free semiring \mathbb{K} over an alphabet Σ [6] is a 5-tuple $(\Sigma, Q, I, F, \delta)$ where Q is a finite set of states and the sets I , F and δ are mappings $I : Q \rightarrow \mathbb{K}$ (input weights), $F : Q \rightarrow \mathbb{K}$ (output weights), and $\delta : Q \times \Sigma \times Q \rightarrow \mathbb{K}$ (transition weights). The set of WFAs on \mathbb{K} is denoted by $\mathcal{M}_{\mathbb{K}}$. A WFA is *homogeneous* if all vertices reaching a same state are labeled by the same letter.

A \mathbb{K} -graph is a graph $G = (X, U)$ labeled with coefficients in \mathbb{K} where X is the set of vertices and $U : X \times X \rightarrow \mathbb{K}$ is the function that associates each edge with its label in \mathbb{K} . When there is no edge from p to q , we have $U(p, q) = \bar{0}$. In case $\mathbb{K} = \mathbb{B}$, the boolean semiring, $E_{\mathbb{B}}$ is the set of *regular expressions* and, as the only element of $\mathbb{K} \setminus \bar{0}$ is $\bar{1}$, we omit the use of coefficient and of the external product ($\bar{1}a = a\bar{1} = a$). For a rational series S represented by $E \in E_{\mathbb{B}}$, $Supp(E)$ is usually called the language of E , denoted by $L(E)$ and $S = Supp(S) = L(E)$. A boolean automaton (automaton in the sequel) M over an alphabet Σ is usually defined [6, 10] as a 5-tuple $(\Sigma, Q, I, F, \delta)$ where Q is a finite set of states, $I \subseteq Q$ the set of initial states, $F \subseteq Q$ the set of final states, and $\delta \subseteq Q \times \Sigma \times Q$ the set of edges. We denote by $L(M)$ the language recognized by the automaton M . A graph $G = (X, U)$ is a \mathbb{B} -graph for which labels of edges are not written.

2.2 Extended Glushkov construction

An algorithm given by Glushkov [8] for computing an automaton with $n + 1$ states from a regular expression of literal length n has been extended to semirings \mathbb{K} by the authors [4]. Informally, the principle is to associate exactly one state in the computed automaton to each occurrence of letters in the expression. Then, we link by a transition two states of the automaton if the two occurrences of the corresponding letters in the expression can be read successively.

In order to recall the extended Glushkov construction, we have to first define the ordered pairs and the supported operations. An ordered pair (l, i) consists of a coefficient $l \in \mathbb{K} \setminus \{\bar{0}\}$ and a position $i \in \mathbb{N}$. We also define the functions $\mathcal{I}_H : H \rightarrow \mathbb{K}$ such that $\mathcal{I}_H(i)$ is equal to $\bar{1}$ if $i \in H$ and $\bar{0}$ otherwise. We define $P : 2^{\mathbb{K} \setminus \{\bar{0}\} \times \mathbb{N}} \rightarrow 2^{\mathbb{N}}$ the function that extracts positions from a set of ordered pairs as follows: for Y a set of ordered pairs, $P(Y) = \{i_j, 1 \leq j \leq |Y| \mid \exists (l_j, i_j) \in Y\}$.

The function $Coeff_Y : P(Y) \rightarrow \mathbb{K} \setminus \{\bar{0}\}$ extracts the coefficient associated to a position i as follows: $Coeff_Y(i) = l$ for $(l, i) \in Y$.

Let $Y, Z \subset \mathbb{K} \setminus \{\bar{0}\} \times \mathbb{N}$ be two sets of ordered pairs. We define the product of $k \in \mathbb{K} \setminus \bar{0}$ and Y by $k \cdot Y = \{(k \otimes l, i) \mid (l, i) \in Y\}$ and $Y \cdot k = \{(l \otimes k, i) \mid (l, i) \in Y\}$, $\bar{0} \cdot Y = Y \cdot \bar{0} = \emptyset$. We define the operation \uplus by $Y \uplus Z = \{(l, i) \mid \text{either } (l, i) \in Y \text{ and } i \notin P(Z) \text{ or } (l, i) \in Z \text{ and } i \notin P(Y) \text{ or } (l_s, i) \in Y, (l_t, i) \in Z \text{ for some } l_s, l_t \in \mathbb{K} \text{ with } l = l_s \oplus l_t \neq \bar{0}\}$.

As in the original Glushkov construction [7, 13], and in order to specify their position in the expression, letters are subscripted following the order of reading. The resulting expression is denoted \bar{E} , defined over the alphabet of indexed symbols $\bar{\Sigma}$, each one appearing at most once in \bar{E} . The set of indices thus obtained is called positions and denoted by $Pos(E)$. For example, starting from $E = (2a+b)^* \cdot a \cdot 3b$, one obtains the indexed expression $\bar{E} = (2a_1 + b_2)^* \cdot a_3 \cdot 3b_4$, $\bar{\Sigma} = \{a_1, b_2, a_3, b_4\}$ and $Pos(E) = \{1, 2, 3, 4\}$. Four functions are defined in order to compute a WFA which needs not be deterministic. $First(E)$ represents the set of initial positions of words of $Supp(\bar{E})$ associated with their input weight, $Last(E)$ represents the set of final positions of words of $Supp(\bar{E})$ associated to their output weight

and $Follow(E, i)$ is the set of positions of words of $Supp(\overline{E})$ which immediately follows position i in the expression \overline{E} , associated to their transition weight. In the boolean case, these sets are subsets of $Pos(E)$. The $Null(E)$ set represents the coefficient of the empty word. The way to compute these sets is completely formalized in table 1.

E	Null(E)	First(E)	Last(E)	Follow(E,i)
\emptyset	$\overline{0}$	\emptyset	\emptyset	\emptyset
ε	$\overline{1}$	\emptyset	\emptyset	\emptyset
a_j	$\overline{0}$	$\{(\overline{1}, j)\}$	$\{(\overline{1}, j)\}$	\emptyset
kF	$k \otimes Null(F)$	$k \cdot First(F)$	$Last(F)$	$Follow(F, i)$
Fk	$Null(F) \otimes k$	$First(F)$	$Last(F) \cdot k$	$Follow(F, i)$
$F + G$	$Null(F) \oplus Null(G)$	$First(F) \uplus First(G)$	$Last(F) \uplus Last(G)$	$\mathcal{I}_{Pos(F)}(i) \cdot Follow(F, i) \uplus \mathcal{I}_{Pos(G)}(i) \cdot Follow(G, i)$
$F \cdot G$	$Null(F) \otimes Null(G)$	$First(F) \uplus Null(F) \cdot First(G)$	$Last(F) \cdot Null(G) \uplus Last(G)$	$\mathcal{I}_{Pos(F)}(i) \cdot Follow(F, i) \uplus \mathcal{I}_{Pos(G)}(i) \cdot Follow(G, i) \uplus Coeff_{Last(F)}(i) \cdot First(G)$
F^+	$\overline{0}$	$First(F)$	$Last(F)$	$Follow(F, i) \uplus Coeff_{Last(F)}(i) \cdot First(F)$
F^*	$\overline{1}$	$First(F)$	$Last(F)$	$Follow(F, i) \uplus Coeff_{Last(F)}(i) \cdot First(F)$

Table 1: Extended Glushkov functions

These functions allow us to define the WFA $\overline{M} = (\overline{\Sigma}, Q, \{s_I\}, F, \overline{\delta})$ where

1. $\overline{\Sigma}$ is the indexed alphabet,
2. s_I is the single initial state with no incoming edge with $\overline{1}$ as input weight,
3. $Q = Pos(E) \cup \{s_I\}$
4. $F : Q \rightarrow \mathbb{K}$ such that $F(i) = \begin{cases} Null(E) & \text{if } i = s_I \\ Coeff_{Last(E)}(i) & \text{otherwise} \end{cases}$
5. $\delta : Q \times \overline{\Sigma} \times Q \rightarrow \mathbb{K}$ such that $\delta(i, a_j, h) = \overline{0}$ for every $h \neq j$, whereas $\delta(i, a_j, j) = \begin{cases} Coeff_{First(E)}(j) & i = s_I \\ Coeff_{Follow(E, i)}(j) & i \neq s_I \end{cases}$

The Glushkov WFA $M = (\Sigma, Q, \{s_I\}, F, \delta)$ of E is computed from \overline{M} by replacing the indexed letters on edges by the corresponding letters in the expression E . We will denote

$A_{\mathbb{K}} : \mathcal{E}_{\mathbb{K}} \rightarrow \mathcal{M}_{\mathbb{K}}$ the application such that $A_{\mathbb{K}}(E)$ is the Glushkov WFA obtained from E by this algorithm proved in [4].

In order to compute a \mathbb{K} -graph from an homogeneous WFA M , we have to add a new vertex $\{\Phi\}$. Then U , the set of edges, is obtained from transitions of M by removing labels and adding directed edges from every final state to $\{\Phi\}$. We label edges to Φ with output weights of final states. The labels of the edges $U(i, p)$ for $i \in Q$, $I(i) \neq \bar{0}$, $p \in Q \cup \{\Phi\}$ are \otimes -multiplied by the input value of the initial state i of M .

In case M is a Glushkov WFA of a \mathbb{K} -expression E , the \mathbb{K} -graph obtained from M is called Glushkov \mathbb{K} -graph of E and is denoted by $G_{\mathbb{K}}(E)$.

2.3 Normal forms and casting operation

Star normal form and epsilon normal form

For the boolean case, Brüggemann-Klein defines regular expressions in *Star Normal Form* (SNF) [2] as expressions E for which, for each position i of $Pos(E)$, when computing the $Follow(E, i)$ function, the unions of sets are disjoint. This definition is given only for usual operators $+$, \cdot , $*$. We can extend this definition to the positive closure, $^+$ as follows:

Definition 1 A \mathbb{B} -expression E is in SNF if, for each closure \mathbb{B} -subexpression H^* or H^+ , the SNF conditions (1) $Follow(H, Last(H)) \cap First(H) = \emptyset$ and (2) $\varepsilon \notin L(H)$ hold.

Then, the properties of the star normal form (defined with the positive closure) are preserved.

In the same paper, Brüggemann-Klein defines also the *epsilon normal form* for the boolean case. We extend this epsilon normal form to the positive closure operator.

Definition 2 The epsilon normal form for a \mathbb{B} -expression E is defined by induction in the following way:

- $[E = \varepsilon \text{ or } E = a]$ E is in epsilon normal form.
- $[E = F + G]$ E is in epsilon normal form if F and G are in epsilon normal form and if $\varepsilon \notin L(F) \cap L(G)$.
- $[E = FG]$ E is in epsilon normal form if F and G are in epsilon normal form.
- $[E = F^+ \text{ or } E = F^*]$ E is in epsilon normal form if F is in epsilon normal form and $\varepsilon \notin L(F)$.

Theorem 3 ([2]) For each regular expression E , there exists a regular expression E^\bullet such that

1. $A_{\mathbb{B}}(E) = A_{\mathbb{B}}(E^\bullet)$,

2. E^\bullet is in SNF
3. E^\bullet can be computed from E in linear time.

Brüggemann-Klein has given every step for the computation of E^\bullet . This computation remains. We just have to add for H^+ the same rules as for H^* . Main steps of the proof are similar.

We extend the star normal form to multiplicities in this way. Let E be a \mathbb{K} -expression. For every subexpression H^* or H^+ in E , for each x in $P(\text{Last}(H))$,

$$P(\text{Follow}(H, x)) \cap P(\text{First}(H)) = \emptyset$$

We do not have to consider the case of the empty word because H^+ and H^* are proper \mathbb{K} -expressions if $c(H) = 0$.

As an example, let $\overline{H} = 2a_1^+ + (3b_2)^+$ and $\overline{E} = (\overline{H})^*$. We can see that the expression $\overline{E} = (2a_1^+ + (3b_2)^+)^*$ is not in SNF, because $2 \in P(\text{Last}(H))$, $2 \in P(\text{Follow}(H, 2)) \cap P(\text{First}(H))$.

The casting operation \sim

We have to define the casting $\sim: \mathcal{M}_{\mathbb{K}} \rightarrow \mathcal{M}_{\mathbb{B}}$. This is similar to the way in which Buchsbaum et al. [3] define the topology of a graph. A WFA $M = (\Sigma, Q, I, F, \delta)$ is casted into an automaton $\widetilde{M} = (\Sigma, Q, \widetilde{I}, \widetilde{F}, \widetilde{\delta})$ in the following way: $\widetilde{I}, \widetilde{F} \subset Q$, $\widetilde{I} = \{q \in Q \mid I(q) \neq \overline{0}\}$, $\widetilde{F} = \{q \in Q \mid F(q) \neq \overline{0}\}$ and $\widetilde{\delta} = \{(p, a, q) \mid p, q \in Q, a \in \Sigma \text{ and } \delta((p, a, q)) \neq \overline{0}\}$. The casting operation can be extended to \mathbb{K} -expressions $\sim: \mathcal{E}_{\mathbb{K}} \rightarrow \mathcal{E}_{\mathbb{B}}$. The regular expression \widetilde{E} is obtained from E by replacing each $k \in \mathbb{K} \setminus \overline{0}$ by $\overline{1}$. The \sim operation on E is an embedding of \mathbb{K} -expressions into regular ones. Nevertheless, the Glushkov \mathbb{B} -graph computed from a \mathbb{K} -expression E may be different whether the Glushkov construction is applied first or the casting operation \sim . This is due to properties of \mathbb{K} -expressions. For example, let $\mathbb{K} = \mathbb{Q}$, $E = 2a^* + (-2)b^*$ (E is not in epsilon normal form). We then have $\widetilde{E} = a^* + b^*$. We can notice that $\widetilde{A}_{\mathbb{K}}(E) \neq A_{\mathbb{B}}(\widetilde{E})$ (E does not recognize ϵ but \widetilde{E} does).

Lemma 4 *Let E be a \mathbb{K} -expression. If E is in SNF and in epsilon normal form, then*

$$\widetilde{A}_{\mathbb{K}}(E) = A_{\mathbb{B}}(\widetilde{E}).$$

Proof We have to show that the automaton obtained by the Glushkov construction for an expression E in $\mathcal{E}_{\mathbb{K}}$ has the same edges as the Glushkov automaton for \widetilde{E} . First, we have $\text{Pos}(E) = \text{Pos}(\widetilde{E})$, as \widetilde{E} is obtained from E only by deleting coefficients. Let us show that $\text{First}(\widetilde{E}) = P(\text{First}(E))$ (states reached from the initial state) by induction on the length of E . If $E = \varepsilon$, $\widetilde{E} = \epsilon$, $\text{First}(\widetilde{E}) = \emptyset = \text{First}(E) = P(\text{First}(E))$. If $E = a \in \Sigma$, $\widetilde{E} = a_1$ then $E = \widetilde{E}$, $\text{First}(E) = \{(\overline{1}, 1)\}$, $P(\text{First}(E)) = \{1\} = \text{First}(\widetilde{E})$.

Let F satisfy the hypothesis, and $E = kF, k \in \mathbb{K} \setminus \bar{0}$. In this case, $\tilde{E} = \tilde{F}$, $P(\text{First}(E)) = P(k.\text{First}(F)) = P(\text{First}(F)) = \text{First}(\tilde{F}) = \text{First}(\tilde{E})$. If $E = Fk$, $k \in \mathbb{K}$, $\tilde{E} = \tilde{F}$, $P(\text{First}(E)) = P(\text{First}(F)) = \text{First}(\tilde{F}) = \text{First}(\tilde{E})$.

If $E = F + H$, and if F and H satisfy the induction hypothesis, and as the coefficient of the empty word is $\bar{0}$ for one of the two subexpression F or H (epsilon normal form), we have $\tilde{E} = \tilde{F} + \tilde{H}$, $\text{First}(\tilde{F} + \tilde{H}) = \text{First}(\tilde{F}) \cup \text{First}(\tilde{H}) = P(\text{First}(F)) \cup P(\text{First}(H))$ which is equal to $P(\text{First}(F + H))$ by induction. We obtain the same result concerning $F \cdot H$, F^+ and F^* .

The equality $\text{Last}(\tilde{E}) = P(\text{Last}(E))$ is obtained similarly.

The last function used to compute the Glushkov automaton is the *Follow* function. Let E be a \mathbb{K} -expression and $i \in \text{Pos}(E)$. If $E = \varepsilon$, $\tilde{E} = \epsilon$, $\text{Follow}(\tilde{E}, i) = \emptyset = \text{Follow}(E, i) = P(\text{Follow}(E, i))$. If $E = a \in \mathbb{K}$, $E = \tilde{E}$, $\text{Follow}(\tilde{E}, i) = \emptyset$. Let F satisfy $\text{Follow}(\tilde{F}, i) = P(\text{Follow}(F, i))$ for all $i \in \text{Pos}(F)$. If E is kF or Fk , $k \in \mathbb{K} \setminus \bar{0}$, $P(\text{Follow}(E, i)) = P(\text{Follow}(F, i)) = \text{Follow}(\tilde{F}, i)$ by hypothesis. If F and H satisfy the induction hypothesis, and if $E = F + H$, (and $i \in \text{Pos}(F)$ without loss of generality), $\text{Follow}(F + H, i) = \text{Follow}(F, i)$, then $P(\text{Follow}(F, i)) = \text{Follow}(\tilde{F}, i)$. We obtain similar results for $E = F.H$ as there is no intersection between positions of F and H . Concerning the star operation, let $E = F^*$, with $\text{Follow}(\tilde{F}, i) = P(\text{Follow}(F, i))$ for all $i \in \text{Pos}(F)$. Then, $P(\text{Follow}(F^*, i)) = P(\text{Follow}(F, i) \cup \text{Coeff}_{\text{Last}(F)}(i) \cdot \text{First}(F))$. But by definition, as F is in SNF, we know that $\text{Follow}(F, i) \cap \text{First}(F) = \emptyset$, so $P(\text{Follow}(F^*, i)) = \text{Follow}(\tilde{F}^*, i)$. In fact, it means that if there exists a couple $(\alpha, j) \in \text{Follow}(F, i)$, there cannot exist $(\beta, j) \in \text{First}(F)$. Otherwise, the expression would not be in SNF, and it would be possible that $\beta = \alpha$, which would make $j \notin \text{Pos}(F^*)$ and imply a deletion of an edge. A same reasoning can be done for the positive closure operator.

Hence, the casting operation \sim and the Glushkov construction commute for the composition operation if we do not consider the empty word. ■

2.4 Characterization of Glushkov automata in the boolean case

The aim of the paper by Caron and Ziadi [5] is to know how boolean Glushkov graphs can be characterized. We recall here the definitions which allow us to give the main theorem of their paper. These notions will be necessary to extend this characterization to Glushkov \mathbb{K} -graphs.

A *hammock* is a graph $G = (X, U)$ without a loop if $|X| = 1$, otherwise it has two distinct vertices i and t such that, for any vertex x of X , (1) there exists a path from i to t going through x , (2) there is no non-trivial path from t to x nor from x to i . Notice that every hammock with at least two vertices has a unique root (the vertex i) and anti-root (the vertex t).

Let $G = (X, U)$ be a hammock. We define $\mathcal{O} = (X_{\mathcal{O}}, U_{\mathcal{O}}) \subseteq G$ as an *orbit* of G if and only if for all x and x' in $X_{\mathcal{O}}$ there exists a non-trivial path from x to x' . The orbit \mathcal{O} is

maximal if, for each vertex $x \in X_{\mathcal{O}}$ and for each vertex $x' \in X \setminus X_{\mathcal{O}}$, there do not exist both a path from x to x' and a path from x' to x . Equivalently, $\mathcal{O} \subseteq G$ is a maximal orbit of G if and only if it is a strongly connected component with at least one edge.

Informally, in a Glushkov graph obtained from a regular expression E , the set of vertices of a maximal orbit corresponds exactly to the set of positions of a closure subexpression of E .

The set of direct successors (respectively direct predecessors) of $x \in X$ is denoted by $Q^+(x)$ (respectively $Q^-(x)$). Let $n_x = |Q^-(x)|$ and $m_x = |Q^+(x)|$. For an orbit $\mathcal{O} \subset G$, $\mathcal{O}^+(x)$ denotes $Q^+(x) \cap (X \setminus \mathcal{O})$ and $\mathcal{O}^-(x)$ denotes the set $Q^-(x) \cap (X \setminus \mathcal{O})$. In other words, $\mathcal{O}^+(x)$ is the set of vertices which are directly reached from x and which are not in \mathcal{O} . By extension, $\mathcal{O}^+ = \bigcup_{x \in \mathcal{O}} \mathcal{O}^+(x)$ and $\mathcal{O}^- = \bigcup_{x \in \mathcal{O}} \mathcal{O}^-(x)$. The sets $In(\mathcal{O}) = \{x \in X_{\mathcal{O}} \mid \mathcal{O}^-(x) \neq \emptyset\}$ and $Out(\mathcal{O}) = \{x \in X_{\mathcal{O}} \mid \mathcal{O}^+(x) \neq \emptyset\}$ denote the *input* and the *output* of the orbit \mathcal{O} . As G is a hammock, $In(\mathcal{O}) \neq \emptyset$ and $Out(\mathcal{O}) \neq \emptyset$. An orbit \mathcal{O} is *stable* if $Out(\mathcal{O}) \times In(\mathcal{O}) \subset U$. An orbit \mathcal{O} is *transverse* if, for all $x, y \in Out(\mathcal{O})$, $\mathcal{O}^+(x) = \mathcal{O}^+(y)$ and, for all $x, y \in In(\mathcal{O})$, $\mathcal{O}^-(x) = \mathcal{O}^-(y)$.

An orbit \mathcal{O} is *strongly stable* (respectively *strongly transverse*) if it is stable (respectively transverse) and if after deleting the edges in $Out(\mathcal{O}) \times In(\mathcal{O})$ (1) there does not exist any suborbit $\mathcal{O}' \subset \mathcal{O}$ or (2) every maximal suborbit of \mathcal{O} is strongly stable (respectively strongly transverse). The hammock G is *strongly stable* (respectively *strongly transverse*) if (1) it has no orbit or (2) every maximal orbit $\mathcal{O} \subset G$ is strongly stable (respectively strongly transverse).

If G is strongly stable, then we call *the graph without orbit* of G , denoted by $SO(G)$, the acyclic directed graph obtained by recursively deleting, for every maximal orbit \mathcal{O} of G , the edges in $Out(\mathcal{O}) \times In(\mathcal{O})$. The graph $SO(G)$ is then reducible if it can be reduced to one vertex by iterated applications of the three following rules:

- **Rule R_1 :** If x and y are vertices such that $Q^-(y) = \{x\}$ and $Q^+(x) = \{y\}$, then delete y and define $Q^+(x) := Q^+(y)$.
- **Rule R_2 :** If x and y are vertices such that $Q^-(x) = Q^-(y)$ and $Q^+(x) = Q^+(y)$, then delete y and any edge connected to y .
- **Rule R_3 :** If x is a vertex such that for all $y \in Q^-(x)$, $Q^+(x) \subset Q^+(y)$, then delete edges in $Q^-(x) \times Q^+(x)$.

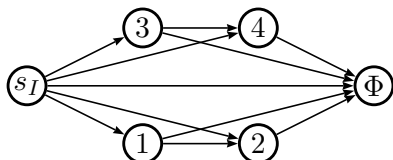
Theorem 5 ([5]) $G = (X, U)$ is a Glushkov graph if and only if the three following conditions are satisfied:

- G is a hammock.
- Each maximal orbit in G is strongly stable and strongly transverse.
- The graph without orbit $SO(G)$ is reducible.

2.5 The problem of reduction rules

An erroneous statement in the paper by Caron and Ziadi

In [5], the definition of the R_3 rules is wrong in some cases. Indeed, if we consider the regular expression $E = (x_1 + \epsilon)(x_2 + \epsilon) + (x_3 + \epsilon)(x_4 + \epsilon)$, the graph obtained from the Glushkov algorithm is as follows



Let us now try to reduce this graph with the reduction rules as they are defined in [5]. We can see that the sequel of applicable rules is R_3 , R_3 and R_1 . We can notice that there is a multiple choice for the application of the first R_3 rule, but after having chosen the vertex on which we will apply this first rule, the sequel of rules leads to a single graph (except with the numerotation of vertices).

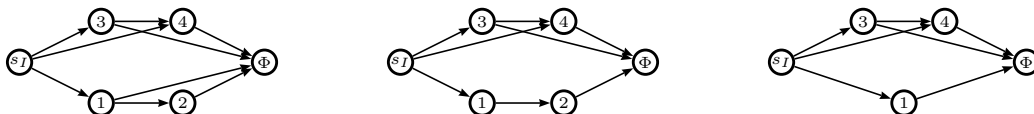


Figure 1: Application of R_3 on 1, R_3 on 2 and R_1 on 1 and 2.

We can see that the graph obtained is no more reducible. This problem is a consequence of the multiple computation of the edge $(0, \Phi)$. In fact, this problem is solved when each edge of the acyclic Glushkov graph is computed only once. It is the case when E is in epsilon normal form.

A new R_3 rule for the boolean case

Let $G = (X, U)$ be an acyclic graph. The rule R_3 is as follows:

- If $x \in X$ is a vertex such that for all $y \in Q^-(x)$, $Q^+(x) \subset Q^+(y)$, then delete the edge $(q^-, q^+) \in Q^-(x) \times Q^+(x)$ if there does not exist a vertex $z \in X \setminus \{x\}$ such that the following conditions are true:
 - there is neither a path from x to z nor a path from z to x ,
 - $q^- \in Q^-(z)$ and $q^+ \in Q^+(z)$,

$$- |Q^-(z)| \times |Q^+(z)| \neq 1.$$

The new rule R_3 check whether conditions of the old R_3 rules are verified and moreover deletes an edge only if it does not correspond to the ε of more than one subexpression. The validity of this rule is shown in Proposition 10.

3 Acyclic Glushkov WFA properties

The definitions of section 2.4 related to graphs are extended to \mathbb{K} -graphs by considering that edges labeled $\bar{0}$ do not exist.

Let us consider M a WFA without orbit. Our aim here is to give conditions on weights in order to check whether M is a Glushkov WFA. Relying on the boolean characterization, we can deduce that M is homogeneous and that the Glushkov graph of \widehat{M} is reducible.

3.1 \mathbb{K} -rules

\mathbb{K} -rules can be seen as an extension of reduction rules. Each rule is divided into two parts: a graphic condition on edges, and a numerical condition (except for the $\mathbb{K}R_1$ -rule) on coefficients. The following definitions allow us to give numerical constraints for the application of \mathbb{K} -rules.

Let $G = (X, U)$ be a \mathbb{K} -graph and let $x, y \in X$. Let us now define the set of beginnings of the set $Q^-(x)$ as $B(Q^-(x)) \subseteq Q^-(x)$. A vertex x^- is in $B(Q^-(x))$ if for all q^- in $Q^-(x)$ there is not a non trivial path from q^- to x^- . In the same way, we define the set of terminations of $Q^+(x)$ as $T(Q^+(x)) \subseteq Q^+(x)$. A vertex x^+ is in $T(Q^+(x))$ if for all q^+ in $Q^+(x)$ there is not a non trivial path from x^+ to q^+ .

We say that x and y are *backward equivalent* if $Q^-(x) = Q^-(y)$ and there exist $l_x, l_y \in \mathbb{K}$ such that for every $q^- \in Q^-(x)$, there exists $\alpha_{q^-} \in \mathbb{K}$ such that $U(q^-, x) = \alpha_{q^-} \otimes l_x$ and $U(q^-, y) = \alpha_{q^-} \otimes l_y$. Similarly, we say that x and y are *forward equivalent* if $Q^+(x) = Q^+(y)$ and there exist $r_x, r_y \in \mathbb{K}$ such that for every $q^+ \in Q^+(x)$, there exists $\beta_{q^+} \in \mathbb{K}$ such that $U(x, q^+) = r_x \otimes \beta_{q^+}$ and $U(y, q^+) = r_y \otimes \beta_{q^+}$. Moreover, if x and y are both backward and forward equivalent, then we say that x and y are *bidirectionally equivalent*.

In the same way, we say that x is *ϵ -equivalent* if for all $(q^-, q^+) \in Q^-(x) \times Q^+(x)$ the edge (q^-, q^+) exists and if there exist $k, l, r \in \mathbb{K}$ such that for every $q^- \in Q^-(x)$ there exists $\alpha_{q^-} \in \mathbb{K}$ and for every $q^+ \in Q^+(x)$ there exist $\beta_{q^+} \in \mathbb{K}$, such that $U(q^-, x) = \alpha_{q^-} \otimes l$, $U(x, q^+) = r \otimes \beta_{q^+}$ and $U(q^-, q^+) = \alpha_{q^-} \otimes k \otimes \beta_{q^+}$.

Similarly, x is *quasi- ϵ -equivalent* if

- $B(Q^-(x)) \neq Q^-(x)$ or $T(Q^+(x)) \neq Q^+(x)$, and
- for all $(q^-, q^+) \in Q^-(x) \times Q^+(x) \setminus B(Q^-(x)) \times T(Q^+(x))$, the edge (q^-, q^+) exists, and

- there exist $k, l, r \in \mathbb{K}$ such that for every $q^- \in Q^-(x)$ there exist $\alpha_{q^-} \in \mathbb{K}$ and for every $q^+ \in Q^+(x)$, there exist $\beta_{q^+} \in \mathbb{K}$ such that $U(q^-, x) = \alpha_{q^-} \otimes l$, $U(x, q^+) = r \otimes \beta_{q^+}$, and
- if $q^- \notin B(Q^-(x))$ or $q^+ \notin T(Q^+(x))$
 - then $U(q^-, q^+) = \alpha_{q^-} \otimes k \otimes \beta_{q^+}$
 - else there exists $\gamma \in \mathbb{K}$ such that $U(q^-, q^+) = \gamma \oplus \alpha_{q^-} \otimes k \otimes \beta_{q^+}$ (Notice that if the edge from q^- to q^+ does not exist in the automaton, then $U(q^-, q^+) = \bar{0}$ and it is possible to have $\gamma \oplus \alpha_{q^-} \otimes k \otimes \beta_{q^+} = \bar{0}$).

In order to clarify our purpose, we have distinguished the case where (q^-, q^+) are superpositions of edges (quasi- ϵ -equivalence of x) to the case where they are not (ϵ -equivalence of x).

Rule $\mathbb{K}R_1$: If x and y are vertices such that $Q^-(y) = \{x\}$ and $Q^+(x) = \{y\}$, then delete y and define $Q^+(x) \leftarrow Q^+(y)$.



Figure 2: $\mathbb{K}R_1$ reduction rule

Rule $\mathbb{K}R_2$: If x and y are bidirectionally equivalent, with $l_x, l_y, r_x, r_y \in \mathbb{K}$ are the constants satisfying such a definition, then

- delete y and any edge connected to y
- for every $q^- \in Q^-(x)$ and $q^+ \in Q^+(x)$ set $U'(q^-, x) = \alpha_{q^-}$ and $U'(x, q^+) = \beta_{q^+}$ where α_{q^-} and β_{q^+} are defined as in the bidirectional equivalence.

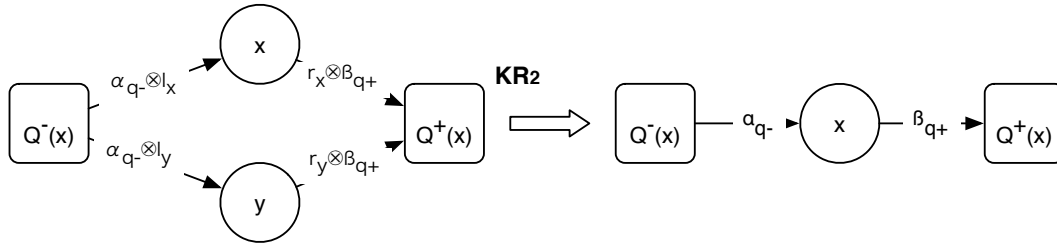


Figure 3: $\mathbb{K}R_2$ reduction rule

Rule $\mathbb{K}R_3$: If x is ϵ -equivalent or x is quasi- ϵ -equivalent with $l, r, k, \gamma \in \mathbb{K}$ the constants satisfying such a definition, then

- if x is ϵ -equivalent
 - then delete every $(q^-, q^+) \in Q^-(x) \times Q^+(x)$,
 - else delete every $(q^-, q^+) \in Q^-(x) \times Q^+(x) \setminus B(Q^-(x)) \times T(Q^+(x))$.
- for every $q^- \in Q^-(x)$ and $q^+ \in Q^+(x)$ set $U'(q^-, x) = \alpha_{q^-}$ and $U'(x, q^+) = \beta_{q^+}$ where α_{q^-} and β_{q^+} are defined as in the ϵ -equivalence or quasi- ϵ -equivalence.
- If x is quasi- ϵ -equivalent then compute the new edges from $B(Q^-(x)) \times T(Q^+(x))$ labeled γ .

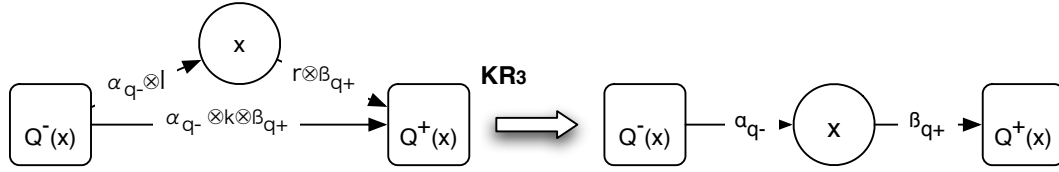


Figure 4: $\mathbb{K}R_3$ reduction when x is ϵ -equivalent

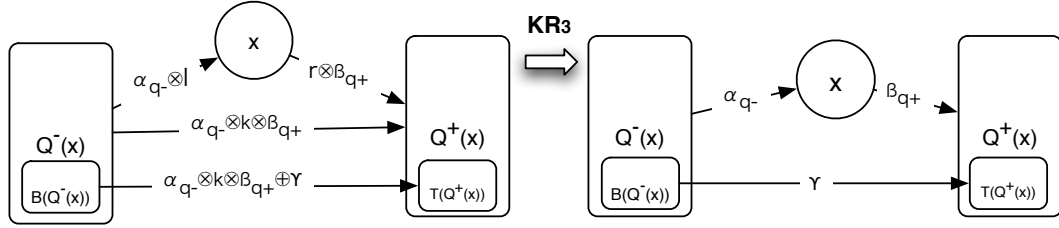


Figure 5: $\mathbb{K}R_3$ reduction when x is quasi- ϵ -equivalent

3.2 Confluence for \mathbb{K} -rules

In order to have an algorithm checking whether a \mathbb{K} -graph is a Glushkov \mathbb{K} -graph, we have to know (1) if it is decidable to apply a \mathbb{K} -rule on some vertices and (2) if the application of \mathbb{K} -rules ends. In order to ensure these characteristics, we will specify some sufficient properties on the semiring \mathbb{K} . Let us define \mathbb{K} as a field or as a factorial semiring. A factorial semiring \mathbb{K} is a zero-divisor free semiring for which every non-zero, non-unit element x of \mathbb{K} can be written as a product of irreducible elements of \mathbb{K} $x = p_1 \cdots p_n$, and this representation is unique apart from the order of the irreducible elements. This notion is a slight adaptation of the factorial ring notion.

It is clear that, if \mathbb{K} is a field, the application of \mathbb{K} -rules is decidable. Conditions of application of \mathbb{K} -rules are sufficient to define an algorithm. In the case of a factorial semiring,

as the decomposition is unique, a gcd is defined¹ and it gives us a procedure allowing us to apply one rule ($\mathbb{K}R_2$ or $\mathbb{K}R_3$) on a \mathbb{K} -graph if it is possible. It ensures the decidability of \mathbb{K} -rules application for factorial semirings. For both cases (field and factorial semiring), we prove that \mathbb{K} -rules are confluent. It ensures the ending of the algorithm allowing us to know whether a \mathbb{K} -graph is a Glushkov one.

We explicit algorithms in order to apply the $\mathbb{K}R_2$ and $\mathbb{K}R_3$ rules. Algorithm 2 tests whether the $\mathbb{K}R_2$ -rule graphical and numerical conditions for two states are verified. If so, it returns the partially reduced \mathbb{K} -graph. Algorithm 1 is divided into three functions. The first one check whether the $\mathbb{K}R_3$ -graphical conditions are checked on a state x ($\mathbb{K}R_3$ GRAPHICALEQUIVALENCECONDITIONSCHECKING) and returns the ϵ or quasi- ϵ -equivalence type of x . Then, depending on the type of x , the numerical conditions for ϵ or quasi- ϵ -equivalence are verified (function EQUIVALENCECHECKING). Finally a partially reduced \mathbb{K} -graph is obtained using GRAPHCOMPUTING function.

Algorithm 1 Application of the $\mathbb{K}R_3$ rule for a state

```

 $\mathbb{K}R_3$ -APPLICATION( $x, G$ )
  ▷ Input: One state  $x$  of a  $\mathbb{K}$ -graph  $G = (X, U)$ 
  ▷ Output: The newly computed graph  $G$ 
  1 Begin
  2   if  $\mathbb{K}R_3$ GRAPHICALEQUIVALENCECONDITIONSCHECKING( $x, G, type$ ) = False then
  3     return False
    ▷ If type is equal to  $\epsilon$  (resp. quasi- $\epsilon$ ) lines labeled {quasi- $\epsilon$ } (resp. { $\epsilon$ })
    ▷ of the functions below are deleted
  4   if EQUIVALENCECHECKING( $x, G, [\alpha], [\beta], k, [\gamma]$ ) = False then
  5     return False
  6   GRAPHCOMPUTING( $x, G, [\alpha], [\beta], k, [\gamma]$ )
  7   return True
  8 End

```

¹In case \mathbb{K} is not commutative, left gcd and right gcd are defined.

Algorithm 2 Application of the $\mathbb{K}R_2$ rule for two states

$\mathbb{K}R_2\text{-APPLICATION}(x, y, G)$

- ▷ **Input:** Two states x and y of a \mathbb{K} -graph $G = (X, U)$
- ▷ **Output:** The newly computed graph G

```
1 Begin
2   if  $Q^-(x) \neq Q^-(y)$  or  $Q^+(x) \neq Q^+(y)$  then
3     return False
4    $q_1^- \leftarrow$  a vertex of  $Q^-(x)$ 
5    $\text{gcd}_r(x) \leftarrow U(q_1^-, x)$ 
6    $\text{gcd}_r(y) \leftarrow U(q_1^-, y)$ 
7   for each  $q^- \in Q^-(x)$  do
8      $\text{gcd}_r(x) \leftarrow \text{RIGHT GCD}(U(q^-, x), \text{gcd}_r(x))$ 
9      $\text{gcd}_r(y) \leftarrow \text{RIGHT GCD}(U(q^-, y), \text{gcd}_r(y))$ 
10  for each  $q^- \in Q^-(x)$  do
11    compute  $\alpha_{q^-}$  such that  $U(q^-, x) = \alpha_{q^-} \otimes \text{gcd}_r(x)$ 
12    if  $\alpha_{q^-} \otimes \text{gcd}_r(y) \neq U(q^-, y)$  then
13      return False
14   $q_1^+ \leftarrow$  a vertex of  $Q^+(x)$ 
15   $\text{gcd}_l(x) \leftarrow U(x, q_1^+)$ 
16   $\text{gcd}_l(y) \leftarrow U(y, q_1^+)$ 
17  for each  $q^+ \in Q^+(x)$  do
18     $\text{gcd}_l(x) \leftarrow \text{LEFT GCD}(U(x, q^+), \text{gcd}_l(x))$ 
19     $\text{gcd}_l(y) \leftarrow \text{LEFT GCD}(U(y, q^+), \text{gcd}_l(y))$ 
20  for each  $q^+ \in Q^+(x)$  do
21    compute  $\beta_{q^+}$  such that  $U(x, q^+) = \text{gcd}_l(x) \otimes \beta_{q^+}$ 
22    if  $\text{gcd}_l(y) \otimes \beta_{q^+} \neq U(y, q^+)$  then
23      return False
24  delete  $y$  and any edge connected to  $y$ 
25  for each  $q^- \in Q^-(x)$  do
26     $U(q^-, x) \leftarrow \alpha_{q^-}$ 
27  for each  $q^+ \in Q^+(x)$  do
28     $U(q^+, x) \leftarrow \beta_{q^+}$ 
29  return True
30 End
```

$\text{GRAPHCOMPUTING}(x, G, [\alpha], [\beta], k, [\gamma])$
 ▷ **Input:** One state x of a \mathbb{K} -graph $G = (X, U)$
 ▷ $\alpha \in \mathbb{K}^{|Q^-(x)|}, \beta \in \mathbb{K}^{|Q^+(x)|}, k \in \mathbb{K}$
 ▷ **Input:** and $\gamma \in \mathbb{K}^{|B(Q^-(x))| \times |T(Q^+(x))|}$ ▷ quasi- ϵ
 ▷ **Output:** The newly computed graph G
 1 **Begin**
 2 **for each** $q^- \in Q^-(x)$ **do**
 3 $U(q^-, x) \leftarrow \alpha_{q^-}$
 4 **for each** $q^+ \in Q^+(x)$ **do**
 5 $U(x, q^+) \leftarrow \beta_{q^+}$
 6 delete any edge $(q^-, q^+) \in Q^-(x) \times Q^+(x)$ ▷ ϵ
 7 delete any edge $(q^-, q^+) \in Q^-(x) \times Q^+(x) \setminus B(Q^-(x)) \times T(Q^+(x))$ ▷ quasi- ϵ
 8 **for each** $(q^-, q^+) \in B(Q^-(x)) \times T(Q^+(x))$ **do** ▷ quasi- ϵ
 9 $U(q^-, q^+) \leftarrow \gamma(q^-, q^+)$ ▷ quasi- ϵ
 10 **End**

$\mathbb{K}R_3\text{GRAPHICALEQUIVALENCECONDITIONSCHECKING}(x, G, \text{type})$
 ▷ **Input:** One state x of a \mathbb{K} -graph $G = (X, U)$
 ▷ **Output:** $\text{type} \in \{\epsilon\text{-equivalence, quasi-}\epsilon\text{-equivalence}\}$
 1 **Begin**
 2 compute $B(Q^-(x))$ and $T(Q^+(x))$
 3 **if** $B(Q^-(x)) = Q^-(x)$ and $T(Q^+(x)) = Q^+(x)$ **then**
 4 **for each** $q^- \in Q^-(x)$ **do**
 5 **for each** $q^+ \in Q^+(x)$ **do**
 6 **if** $U(q^-, q^+) = \bar{0}$ **then**
 7 **return** *False*
 8 $\text{type} \leftarrow \epsilon$
 9 **return** *True*
 10 **else for each** $q^- \in Q^-(x)$ **do**
 11 **for each** $q^+ \in Q^+(x)$ **do**
 12 **if** $(q^-, q^+) \in Q^-(x) \times Q^+(x) \setminus B(Q^-(x)) \times T(Q^+(x))$
 and $U(q^-, q^+) = \bar{0}$ **then**
 13 **return** *False*
 14 $\text{type} \leftarrow \text{quasi-}\epsilon$
 15 **return** *True*
 16 **End**


```

EQUIVALENCHECKING( $x, G, [\alpha], [\beta], k, [\gamma]$ )
  ▷ Input: One state  $x$  of a  $\mathbb{K}$ -graph  $G = (X, U)$ 
  ▷ Output:  $\alpha \in \mathbb{K}^{|Q^-(x)|}, \beta \in \mathbb{K}^{|Q^+(x)|}, k \in \mathbb{K}$ 
  ▷ Output: and  $\gamma \in \mathbb{K}^{|B(Q^-(x))| \times |T(Q^+(x))|}$  ▷ quasi- $\epsilon$ 
1 Begin
2    $q_1^- \leftarrow$  a vertex of  $Q^-(x)$ 
3    $\text{gcd}_r \leftarrow U(q_1^-, x)$ 
4   for each  $q^- \in Q^-(x)$  do
5      $\text{gcd}_r \leftarrow \text{RIGHT GCD}(U(q^-, x), \text{gcd}_r)$ 
6   for each  $q^- \in Q^-(x)$  do
7     compute  $\alpha_{q^-}$  such that  $U(q^-, x) = \alpha_{q^-} \otimes \text{gcd}_r$ 
8    $q_1^+ \leftarrow$  a vertex of  $Q^+(x)$ 
9    $\text{gcd}_l \leftarrow U(x, q_1^+)$ 
10  for each  $q^+ \in Q^+(x)$  do
11     $\text{gcd}_l \leftarrow \text{LEFT GCD}(\text{gcd}_l, U(x, q^+))$ 
12  for each  $q^+ \in Q^+(x)$  do
13    compute  $\beta_{q^+}$  such that  $U(x, q^+) = \text{gcd}_l \otimes \beta_{q^+}$ 
14   $(q_1^-, q_1^+) \leftarrow$  a couple of vertices of  $Q^-(x) \times Q^+(x)$  ▷  $\epsilon$ 
15   $(q_1^-, q_1^+) \leftarrow$  a couple of vertices of
     $Q^-(x) \times Q^+(x) \setminus B(Q^-(x)) \times T(Q^+(x))$  ▷ quasi- $\epsilon$ 
16  Find  $k_1$  such that
     $U(q_1^-, q_1^+) = \alpha_{q_1^-} \otimes k_1 \otimes \beta_{q_1^+}$ 
17  if  $k_1$  does not exist then
18    return False
19  for each  $(q^-, q^+) \in Q^-(x) \times Q^+(x)$  do
20    if  $(q^-, q^+) \notin B(Q^-(x)) \times T(Q^+(x))$  then ▷ quasi- $\epsilon$ 
21      Find  $k$  such that
         $U(q^-, q^+) = \alpha_{q^-} \otimes k \otimes \beta_{q^+}$ 
22      if  $k$  does not exist then
23        return False
24      elif  $k \neq k_1$  then
25        return False
26  for each  $(q^-, q^+) \in B(Q^-(x)) \times T(Q^+(x))$  do ▷ quasi- $\epsilon$ 
27    Find  $\gamma(q^-, q^+)$  such that ▷ quasi- $\epsilon$ 
         $U(q^-, q^+) = \alpha_{q^-} \otimes k \otimes \beta_{q^+} \oplus \gamma(q^-, q^+)$ 
28    if  $\gamma(q^-, q^+)$  does not exist then ▷ quasi- $\epsilon$ 
29      return False ▷ quasi- $\epsilon$ 
30  return True
31 End

```

Definition 6 (confluence) Let G be a \mathbb{K} -graph and \mathbb{I}_G the acyclic graph having only one vertex. Let R_1 be a sequence of \mathbb{K} -rules such that

$$G \xrightarrow{R_1} \mathbb{I}_G$$

\mathbb{K} -rules are confluent if for all \mathbb{K} -graph G_2 such that there exists R_2 a sequence of \mathbb{K} -rules with $G \xrightarrow{R_2} G_2$ then there exists R'_2 a sequence of \mathbb{K} -rules such that

$$G_2 \xrightarrow{R'_2} \mathbb{I}_G$$

For the following, \mathbb{K} is a field or a factorial semiring.

Proposition 7 *The \mathbb{K} -rules are confluent.*

Proof In order to prove this result, we will show that if there exist two applicable \mathbb{K} -rules reducing a Glushkov \mathbb{K} -graph, then the order of application does not modify the resulting \mathbb{K} -graph.

Let us denote by $r_{x,y}(G)$ the application of a $\mathbb{K}R_1$, $\mathbb{K}R_2$ or $\mathbb{K}R_3$ rule on the vertices x and y with $y = \emptyset$ for a $\mathbb{K}R_3$ rule.

Let $G = (X, U)$ be a Glushkov \mathbb{K} -graph and let $r_{x,y}$ and $r_{z,t}$ be two applicable \mathbb{K} -rules on G such that $\{x, y\} \cap \{z, t\} = \emptyset$ and no edge can be deleted by both rules. Necessarily we have $r_{x,y}(r_{z,t}(G)) = r_{z,t}(r_{x,y}(G))$.

Suppose now that $\{x, y\} \cap \{z, t\} \neq \emptyset$ or one edge is deleted by both rules. We have to consider several cases depending on the rule $r_{x,y}$.

$r_{x,y}$ is a $\mathbb{K}R_1$ rule In this case $r_{z,t}$ can not delete the edge from x to y and $r_{z,t}$ is necessarily a $\mathbb{K}R_1$ -rule with $\{x, y\} \cap \{z, t\} \neq \emptyset$. If $y = z$, as the coefficient does not act on the reduction rule, $r_{x,y}(r_{z,t}(G)) = r_{z,t}(r_{x,y}(G))$

$r_{x,y}$ is a $\mathbb{K}R_2$ rule Consider that $r_{z,t}$ is a $\mathbb{K}R_2$ rule with $y = z$. Using the notations of the $\mathbb{K}R_2$ rule, there exist α_{q^-} , β_{q^+} , l_x, l_y, r_x, r_y such that $U(q^-, x) = \alpha_{q^-} l_x$, $U(q^-, y) = \alpha_{q^-} l_y$, $U(x, q^+) = r_x \beta_{q^+}$ and $U(y, q^+) = r_y \beta_{q^+}$ with $q^- \in Q^-(x)$, $q^+ \in Q^+(x)$, and $l_x = \gcd_r(x)$, $l_y = \gcd_r(y)$ ($r_x = \gcd_l(x)$, $r_y = \gcd_l(y)$). By hypothesis, a $\mathbb{K}R_2$ rule can also be applied on the vertices y and t . There also exists α'_{q^-} , β'_{q^+} , l'_x, l'_t, r'_x, r'_t such that $\alpha_{q^-} = \alpha'_{q^-} l'_x$, $\beta_{q^+} = r'_x \beta'_{q^+}$, $U(q^-, t) = \alpha'_{q^-} l'_t$, $U(t, q^+) = r'_t \beta'_{q^+}$ ($Q^-(x) = Q^-(t)$ and $Q^+(x) = Q^+(t)$). By construction (Algorithm 2) of $\gcd_r(x)$, the left gcd of all α_{q^-} is $\bar{1}$. Then, whatever the order of application of $\mathbb{K}R_2$ rules, the same decomposition of edges values is obtained. Symetrically a same reasoning is applied for the right part.

Consider now that $r_{z,t} = r_{z,\emptyset}$ is a $\mathbb{K}R_3$ rule. Neither edges from x or y nor edges to x or y can be deleted by $r_{z,\emptyset}$. Then $z = x$ or $z = y$. Let $z = y$. If we successively apply $r_{x,y}$ and $r_{y,\emptyset}$ or $r_{y,\emptyset}$ and $r_{x,y}$ on G , we obtain the same \mathbb{K} -graph following the same method (function EQUIVALENCECHECKING) as the previous case. If we choose $z = x$, we have also the same \mathbb{K} -graph (commutativity property of the sum operator).

$r_{x,\emptyset}$ is a $\mathbb{K}R_3$ rule The only case to consider now is $r_{z,t} = r_{z,\emptyset}$ a $\mathbb{K}R_3$ rule. Suppose that $r_{z,\emptyset}$ deletes an edge also deleted by $r_{x,\emptyset}$ (with $x \neq z$). Let (q^-, q^+) be this edge.

Using the notations of the $\mathbb{K}R_3$ rule, there exist $\alpha_{q^-}, \beta_{q^+}, l, r$ such that $U(q^-, x) = \alpha_{q^-}l$, $U(x, q^+) = r\beta_{q^+}$, $U(q^-, q^+) = \alpha_{q^-}k\beta_{q^+} \oplus \gamma$ with $q^- \in Q^-(x)$, $q^+ \in Q^+(x)$ and $l = \gcd_r(x)$, $r = \gcd_l(x)$. There also exists $\alpha'_{q^-}, \beta'_{q^+}, l', r'$ such that $U(q^-, z) = \alpha'_{q^-}l'$, $U(z, q^+) = r'\beta'_{q^+}$, $U(q^-, q^+) = \alpha'_{q^-}k'\beta'_{q^+} \oplus \gamma'$ with $l' = \gcd_{r'}(z)$, $r' = \gcd_{l'}(z)$. By construction, (function `EQUIVALENCECHECKING`), the computation of l and l' (r and r') are independant. A same reasoning is applied for the right part. Then we can choose γ'' such that $\gamma = \alpha'_{q^-}k'\beta'_{q^+} \oplus \gamma''$ and $\gamma' = \alpha_{q^-}k\beta_{q^+} \oplus \gamma''$. So $U(q^-, q^+) = \alpha_{q^-}k\beta_{q^+} \oplus \alpha'_{q^-}k'\beta'_{q^+} \oplus \gamma''$. It is easy to see that $r_{x,\emptyset}(r_{z,\emptyset}(G)) = r_{z,\emptyset}(r_{x,\emptyset}(G))$.

■

3.3 \mathbb{K} -reducibility

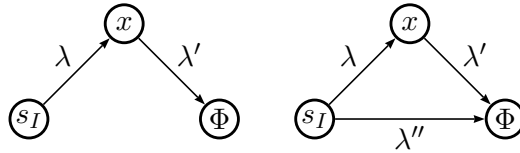
Definition 8 A \mathbb{K} -graph $G = (X, U)$ is said to be \mathbb{K} -reducible if it has no orbit and if it can be reduced to one vertex by iterated applications of any of the three rules $\mathbb{K}R_1$, $\mathbb{K}R_2$, $\mathbb{K}R_3$ described below.

Proposition 10 shows the existence of a sequel of \mathbb{K} -rules leading to the complete reduction of Glushkov \mathbb{K} -graphs. However, the existence of an algorithm allowing us to obtain this sequel of \mathbb{K} -rules depends on the semiring \mathbb{K} .

In order to show the \mathbb{K} -reducibility property of a Glushkov \mathbb{K} -graph G , we check (Lemma 9) that every sequence \mathcal{R} of \mathbb{K} -rules leading to the \mathbb{K} -reduction of G contains necessarily two $\mathbb{K}R_1$ rules which will be denoted by r_\circ and r_\bullet .

Lemma 9 Let $G = (X, U)$ be a \mathbb{K} -reducible Glushkov \mathbb{K} -graph without orbit with $|X| \geq 3$, and let $\mathcal{R} = r_1 \cdots r_n$ be the sequence of \mathbb{K} -rules which can be applied on G and reduce it. Necessarily, \mathcal{R} can be written $\mathcal{R}'r_\circ r_\bullet$ with r_\circ and r_\bullet two $\mathbb{K}R_1$ -rules merging respectively s_I and Φ .

Proof We show this lemma by induction on the number of vertices of the graph. It is obvious that if $|X| = 3$ then, the only possible graphs are the following ones:



and then, for the first one $\mathcal{R} = r_\circ r_\bullet$ with $k = \lambda$ in r_\circ and $k = \lambda'$ in r_\bullet . For the second one x is ϵ -equivalent and $\mathcal{R} = rr_\circ r_\bullet$ with r a $\mathbb{K}R_3$ -rule such that $\alpha = \bar{1}$, $\beta = \bar{1}$, $l = \lambda$, $r = \lambda'$ and $k = \lambda''$. Then, r_\circ and r_\bullet are $\mathbb{K}R_1$ rules such that $k = \bar{1}$ for r_\circ and r_\bullet . Suppose now that G has n vertices. As it is \mathbb{K} -reducible, there exists a sequence of \mathbb{K} -rules which leads to one of the two previous basic cases.

■

For the reduction process, we associate each vertex of G to a subexpression. We define $E(x)$ to be the expression of the vertex x . At the beginning of the process, $E(x)$ is a , the only letter labelling edges reaching the vertex x (homogeneity of Glushkov automata). For the vertices s_I and Φ , we define $E(s_I) = E(\Phi) = \epsilon$. When applying \mathbb{K} -rules, we associate a new expression to each new vertex. With notations of figure 2, the $\mathbb{K}R_1$ -rule induces $E(x) \leftarrow E(x) \cdot k \times E(y)$ with $k = U(x, y)$. With notations of figure 3, the $\mathbb{K}R_2$ -rule induces $E(x) \leftarrow l_x E(x) r_x + l_y E(y) r_y$. And with notations of figures 4 and 5, the $\mathbb{K}R_3$ -rule induces $E(x) \leftarrow l E(x) r + k$.

Proposition 10 *Let $G = (X, U)$ be a \mathbb{K} -graph without orbit. The graph G is a Glushkov \mathbb{K} -graph if and only if it is \mathbb{K} -reducible.*

(\Rightarrow) This proposition will be proved by recurrence on the length of the expression. First for $\|E\| = 1$, we have only two proper \mathbb{K} -expressions which are $E = \lambda$ and $E = \lambda a \lambda'$, for $\lambda, \lambda' \in \mathbb{K}$. When $E = \lambda$, the Glushkov \mathbb{K} -graph has only two vertices which are s_I and Φ and the edge (s_I, Φ) is labeled with λ . Then the $\mathbb{K}R_1$ rule can be applied. Suppose now that $E = \lambda a \lambda'$, then the Glushkov \mathbb{K} -graph of E has three vertices and is \mathbb{K} -reducible. Indeed, the $\mathbb{K}R_1$ -rule can be applied twice.

Suppose now that for each proper \mathbb{K} -expression E of length n , its Glushkov \mathbb{K} -graph is \mathbb{K} -reducible. We then have to show that the Glushkov \mathbb{K} -graph of \mathbb{K} -expressions $F = E + \lambda$, $F = E + \lambda a \lambda'$, $F = \lambda a \lambda' \cdot E$ and $F = E \cdot \lambda a \lambda'$ of length $n + 1$ are \mathbb{K} -reducible. Let us denote by \mathcal{R} (respectively \mathcal{R}') the sequence of rules which can be applied on $A_{\mathbb{K}}(E)$ (respectively $A_{\mathbb{K}}(F)$). In case $|X| \geq 3$, $\mathcal{R} = \mathcal{R}_b r \circ r_\bullet$ (respectively $\mathcal{R}' = \mathcal{R}'_b r'_\bullet r'_\bullet$).

case $F = E + \lambda$ We have $Pos(F) = Pos(E)$, $First(F) = First(E)$, $Last(F) = Last(E)$, $Null(F) = Null(E) + \lambda$ and $\forall i \in Pos(E)$, $Follow(F, i) = Follow(E, i)$. Every rule which can be applied on $A_{\mathbb{K}}(E)$ and which does not modify the edge (s_I, Φ) can also be applied on $A_{\mathbb{K}}(F)$.

If $A_{\mathbb{K}}(E)$ has only two states, then $\mathcal{R} = r$ a $\mathbb{K}R_1$ -rule, and then $\mathcal{R}' = r'$ a $\mathbb{K}R_1$ -rule where r' is such that $k = Null(E) + \lambda$. Elsewhere, the (s_I, Φ) edge can only be reduced by a $\mathbb{K}R_3$ rule.

Suppose now that there is no $\mathbb{K}R_3$ rule modifying (s_I, Φ) which can be applied on $A_{\mathbb{K}}(E)$. Then there is a $\mathbb{K}R_3$ rule r' which can be applied on $A_{\mathbb{K}}(F)$ with $k = \lambda$ and then $A_{\mathbb{K}}(F)$ can be reduced by $\mathcal{R}' = \mathcal{R}_b r' r'_\bullet r'_\bullet$.

Let us now suppose that r_1, r_2, \dots, r_n is the subsequence of $\mathbb{K}R_3$ -rules of \mathcal{R} which modify the (s_I, Φ) edge. Necessarily, r_n acts on a state x which is ϵ -equivalent. If $Q^-(x) \neq \{s_I\}$ or $Q^+(x) \neq \{\Phi\}$ then $\mathcal{R}'_b = \mathcal{R}_b r_{n+1}$ where r_n in \mathcal{R}'_b is modified as follows: x is quasi- ϵ -equivalent with $\gamma = \lambda$ and the rule r_{n+1} is a $\mathbb{K}R_3$ rule on a state x which is ϵ -equivalent and $k = \lambda$. Elsewhere, there is two cases to distinguish. If $Null(E) \oplus \lambda = \bar{0}$ then the r_n rule is no more applicable on $A_{\mathbb{K}}(F)$ (no edge between

s_I and Φ) and the r_{n-1} rule in \mathcal{R}' now acts on an ϵ -equivalent vertex in $A_{\mathbb{K}}(F)$. If $Null(E) + \lambda \neq \bar{0}$ then r_n can be applied on $A_{\mathbb{K}}(F)$ with $k = k \oplus \lambda$.

case $F = E + \lambda a \lambda'$ If $|Pos(E)| = n$, we have, $Pos(F) = Pos(E) \cup \{n+1\}$, $First(F) = First(E) \uplus \{(\lambda, n+1)\}$, $Last(F) = Last(E) \uplus \{(\lambda', n+1)\}$, $Null(F) = Null(E)$ and $\forall i \in Pos(E)$, $Follow(F, i) = Follow(E, i)$ and $Follow(F, n+1) = \emptyset$. In this case, $\mathcal{R}' = \mathcal{R}_b r r'_{\circ} r'_{\bullet}$ where r is a $\mathbb{K}R_2$ rule with $\alpha_{s_I} = \beta_{\Phi} = \bar{1}$ and $l_y = \lambda$, $r_y = \lambda'$ and so $A_{\mathbb{K}}(F)$ is \mathbb{K} -reducible.

case $F = E \cdot \lambda a \lambda'$ If $|Pos(E)| = n$, we have, $Pos(F) = Pos(E) \cup \{n+1\}$, $First(F) = First(E)$, $Last(F) = \{(\lambda', n+1)\}$, $Null(F) = \emptyset$ and $\forall i \in Pos(E) \setminus P(Last(E))$, $Follow(F, i) = Follow(E, i)$ and $\forall i \in P(Last(E))$, $Follow(F, i) = Follow(E, i) \uplus \{(\lambda, n+1)\}$. Let r_1, \dots, r_n be the subsequence of \mathbb{K} -rules modifying edges reaching Φ . Necessarily, $n = 1$ and $r_1 = r_{\bullet}$ (Lemma 9). Indeed, let us suppose that $n > 1$ and that there exists $j \neq i$ such that r_j is a $\mathbb{K}R_1$, $\mathbb{K}R_2$, or $\mathbb{K}R_3$ -rule. Necessarily $|Q^-(\Phi)| \geq 1$, which contradicts our hypothesis. Then we have $\mathcal{R}' = \mathcal{R} r_{n+1}$ where r_{\bullet} the $\mathbb{K}R_1$ -rule from a vertex x to Φ of the sequence \mathcal{R} and labeled with k_i is modified in \mathcal{R}' as follows: $k = k_i \otimes \lambda$. We have also $k = \lambda'$ for the rule r_{n+1} .

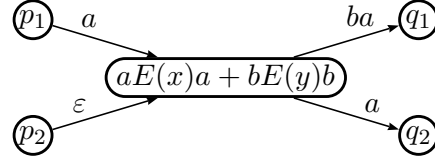
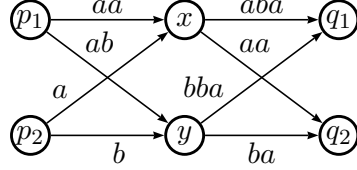
The case $F = \lambda a \lambda' \cdot E$ is proved similarly as the previous one considering the rules modifying edges from s_I (with r_{\circ} instead of r_{\bullet}).

(\Leftarrow) By induction on the number of states of the reducible \mathbb{K} -graph $G = (X, U)$. If $|X| = 2$, $X = \{s_I, \Phi\}$ and the only \mathbb{K} -expression E is λ with $\lambda \in \mathbb{K}$. Let $G' = (X', U')$ be the Glushkov \mathbb{K} -graph obtained from E . By construction $\lambda = U(s_I, \Phi) = E(s_I)$ and $\lambda = Null(E)$, necessarily $G' = G$.

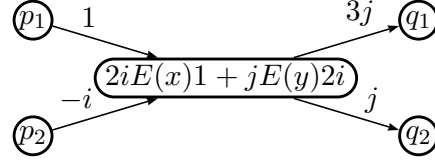
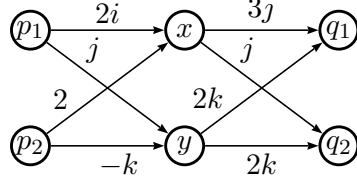
We consider the property true for ranks bellow $n+1$ and G a \mathbb{K} -graph partially reduced. Three cases can occur according to the graphic form of the partially reduced graph. Either we will have to apply twice the $\mathbb{K}R_1$ -rule or once the $\mathbb{K}R_3$ -rule and twice the $\mathbb{K}R_1$ -rule if $X = \{s_I, x, \Phi\}$, or we will have to apply once the $\mathbb{K}R_2$ -rule and twice the $\mathbb{K}R_1$ -rule if $X = \{s_I, x, y, \Phi\}$. For each case, we compute successively the new expressions of vertices, and we check that the Glushkov construction applied on the final \mathbb{K} -expression is G . ■

3.4 Several examples of use for \mathbb{K} -rules

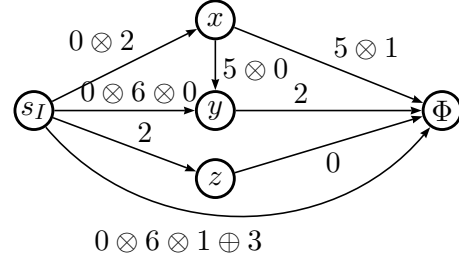
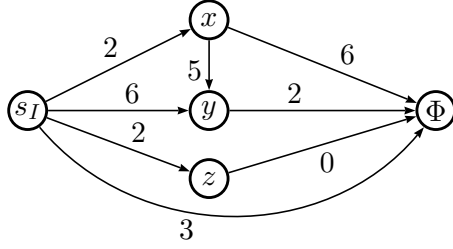
For the $\mathbb{K}R_2$ rule, the first example is for transducers in $(\mathbb{K}, \oplus, \otimes) = (\Sigma^* \cup \emptyset, \cup, \cdot)$ where “ \cdot ” denotes the concatenation operator. In this case, we can express the $\mathbb{K}R_2$ rule conditions as follows. For all q^- in $Q^-(x)$, α_{q^-} is the common prefix of $U(q^-, x)$ and $U(q^-, y)$. Likewise, for all q^+ in $Q^+(x)$, β_{q^+} is the common suffix of $U(q^+, x)$ and $U(q^+, y)$.



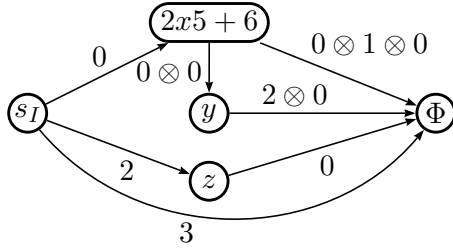
The second one is in $(\mathbb{Z}/7\mathbb{Z}[i, j, k], \oplus, \otimes)$, where $\{i, j, k\}$ are elements of the quaternions and \oplus is the sum and \otimes the product. In this case, \mathbb{K} is a field. Every factorization leads to the result.



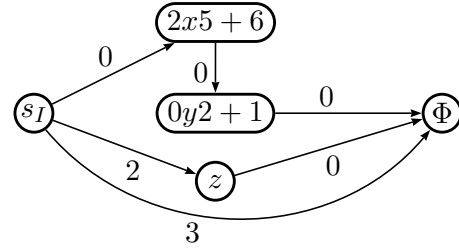
We now give a complete example using the three rules on the $(\mathbb{N} \cup \{+\infty\}, \min, +)$ semiring. This example enlightens the reader on the problem of the quasi-epsilon equivalence. For this example, we will identify the vertex with its label.



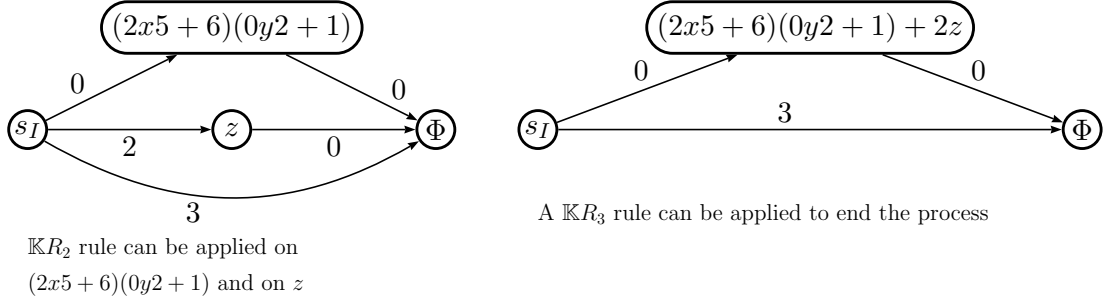
$\mathbb{K}R_3$ rule can be applied on x with $l = 2$, $r = 5$ and $k = 6$



$\mathbb{K}R_3$ rule can be applied on y with $l = 0$, $r = 2$ and $k = 1$



$\mathbb{K}R_1$ rule can be applied on $(2x5 + 6)$ and on $(0y2 + 1)$



This example leads to a possible \mathbb{K} -expression such as $E = ((2x5 + 6)(0y2 + 1) + 2z) + 3$

4 Glushkov \mathbb{K} -graph with orbits

We will now consider a graph which has at least one maximal orbit \mathcal{O} . We extend the notions of strong stability and strong transversality to the \mathbb{K} -graphs obtained from \mathbb{K} -expressions in SNF. We have to give a characterization on coefficients only. The stability and transversality notions are rather linked. Indeed, if we consider the states of $In(\mathcal{O})$ as those of \mathcal{O}^+ then both notions amount to the transversality. Moreover, the extension of these notions to WFAs (\mathbb{K} -stability - definition 12 - and \mathbb{K} -transversality - definition 14), implies the manipulation of output and input vectors of \mathcal{O} whose product is exactly the orbit matrix of \mathcal{O} (Proposition 17).

Lemma 11 *Let E be a \mathbb{K} -expression and $G_{\mathbb{K}}(E)$ its Glushkov \mathbb{K} -graph. Let $\mathcal{O} = (X_{\mathcal{O}}, U_{\mathcal{O}})$ be a maximal orbit of $G_{\mathbb{K}}(E)$. Then E contains a closure subexpression F such that $X_{\mathcal{O}} = Pos(F)$.*

This lemma is a direct consequence of Lemma 4.5 in [5] and of Lemma 4.

Definition 12 (\mathbb{K} -stability) *A maximal orbit \mathcal{O} of a \mathbb{K} -graph $G = (X, U)$ is \mathbb{K} -stable if*

- $\tilde{\mathcal{O}}$ is stable and
- the matrix $M_{\mathcal{O}} \in \mathbb{K}^{|Out(\mathcal{O})| \times |In(\mathcal{O})|}$ such that $M_{\mathcal{O}}(s, e) = U(s, e)$, for each (s, e) of $Out(\mathcal{O}) \times In(\mathcal{O})$, can be written as a product VW of two vectors such that $V \in \mathbb{K}^{|Out(\mathcal{O})| \times 1}$ and $W \in \mathbb{K}^{1 \times |In(\mathcal{O})|}$.

The graph G is \mathbb{K} -stable if each of its maximal orbits is \mathbb{K} -stable.

If a maximal orbit \mathcal{O} is \mathbb{K} -stable, $M_{\mathcal{O}}$ is a matrix of rank 1 called the *orbit matrix*. Then, for a decomposition of $M_{\mathcal{O}}$ in the product VW of two vectors, V will be called the *tail-orbit vector* of \mathcal{O} and W will be called the *head-orbit vector* of \mathcal{O} .

Lemma 13 *A Glushkov \mathbb{K} -graph obtained from a \mathbb{K} -expression E in SNF is \mathbb{K} -stable.*

Proof Let G be the Glushkov \mathbb{K} -graph of a \mathbb{K} -expression E in SNF, $A_{\mathbb{K}}(E) = (\Sigma, Q, s_I, F, \delta)$ its Glushkov WFA and $\mathcal{O} = (X_{\mathcal{O}}, U_{\mathcal{O}})$ be a maximal orbit of G . Following Lemma 4 and Theorem 5, G is strongly stable which implies that every orbit of G is stable. Let $s_i \in \text{Out}(\mathcal{O})$, $1 \leq i \leq |\text{Out}(\mathcal{O})|$ and $e_j \in \text{In}(\mathcal{O})$, $1 \leq j \leq |\text{In}(\mathcal{O})|$. Following the extended Glushkov construction and as for all $s_i \in \text{Out}(\mathcal{O})$, $s_i \neq s_I$, we have $\delta(s_i, a, e_j) = \text{Coeff}_{\text{Follow}(E, s_i)}(e_j)$. As \mathcal{O} corresponds to a closure subexpression F^* or F^+ (Lemma 11) and as (s_i, a, e_j) is an edge of $X_{\mathcal{O}} \times \Sigma \times X_{\mathcal{O}}$, we have $\delta(s_i, a, e_j) = \text{Coeff}_{\text{Follow}(F^*, s_i)}(e_j) = \text{Coeff}_{\text{Follow}(F, s_i) \uplus \text{Coeff}_{\text{Last}(F)}(s_i) \cdot \text{First}(F)}(e_j)$. As E is in SNF, so are F^* and F^+ , and then $\delta(s_i, a, e_j) = \text{Coeff}_{\text{Coeff}_{\text{Last}(F)}(s_i) \cdot \text{First}(F)}(e_j) = \text{Coeff}_{\text{Last}(F)}(s_i) \cdot \text{Coeff}_{\text{First}(F)}(e_j)$. The lemma is proved choosing $V \in \mathbb{K}^{|\text{Out}(\mathcal{O})| \times 1}$ such that $V(i, 1) = \text{Coeff}_{\text{Last}(F)}(s_i)$ and $W \in \mathbb{K}^{1 \times |\text{In}(\mathcal{O})|}$ with $W(1, j) = \text{Coeff}_{\text{First}(F)}(e_j)$. ■

Definition 14 (\mathbb{K} -transversality) *A maximal orbit \mathcal{O} of $G = (X, U)$ is \mathbb{K} -transverse if*

- $\tilde{\mathcal{O}}$ is transverse,
- the matrix $M_e \in \mathbb{K}^{|\mathcal{O}^-| \times |\text{In}(\mathcal{O})|}$ such that $M_e(p, e) = U(p, e)$ for each (p, e) of $\mathcal{O}^- \times \text{In}(\mathcal{O})$, can be written as a product ZT of two vectors such that $Z \in \mathbb{K}^{|\mathcal{O}^-| \times 1}$ and $T \in \mathbb{K}^{1 \times |\text{In}(\mathcal{O})|}$,
- the matrix $M_s \in \mathbb{K}^{|\text{Out}(\mathcal{O})| \times |\mathcal{O}^+|}$ such that $M_s(s, q) = U(s, q)$ for each (s, q) of $\text{Out}(\mathcal{O}) \times \mathcal{O}^+$, can be written as a product $T'Z'$ of two vectors such that $T' \in \mathbb{K}^{|\text{Out}(\mathcal{O})| \times 1}$ and $Z' \in \mathbb{K}^{1 \times |\mathcal{O}^+|}$.

The graph G is \mathbb{K} -transverse if each of its maximal orbits is \mathbb{K} -transverse.

If a maximal orbit \mathcal{O} is \mathbb{K} -transverse, M_e (respectively M_s) is a matrix of rank 1 called the *input matrix* of \mathcal{O} (respectively *output matrix* of \mathcal{O}). For a decomposition of M_e (respectively M_s) in the product ZT (respectively $T'Z'$) of two vectors, T will be called the *input vector* (respectively T' will be called the *output vector*) of \mathcal{O} .

Lemma 15 *The Glushkov \mathbb{K} -graph $G = (X, U)$ of a \mathbb{K} -expression E in SNF is \mathbb{K} -transverse.*

Proof Let \mathcal{O} be a maximal orbit of G . Following Lemma 4 and Theorem 5, G is strongly transverse implies that \mathcal{O} is transverse. By Lemma 11, there exists a maximal closure subexpression H such that $H = F^*$ or $H = F^+$. As E is in SNF, so is H . By the definition of the function *Follow*, we have in this case: for all $p \in \text{Out}(\mathcal{O})$, for all $q \in \mathcal{O}^+$, $U(p, q) = \text{Coeff}_{\text{Follow}(F, p)}(q)$. We now have to distinguish three cases.

1. If $|\mathcal{O}^+| = 1$, then the result holds immediatly. Indeed the output matrix of \mathcal{O} is a vector.
2. If $\mathcal{O}^+ = \{q_1, \dots, q_n\}$ and $n > 1$, $\forall 1 \leq j \leq n, q_j \neq \Phi$, necessarily, we have $\mathcal{O}^+ = \bigcup_l P(\text{First}(H_l))$ with H_l some subexpressions of E . Then we have $U(p, q_j) = \text{Coeff}_{\text{Coeff}_{\text{Last}(F)}(p), \text{First}(H_l)}(q_j)$ if $q_j \in P(\text{First}(H_l))$. Then as q_j is a first position of only one subexpression, $U(p, q_j) = k_p \otimes \text{Coeff}_{\text{First}(H_l)}(q_j)$ where $k_p = \text{Coeff}_{\text{Last}(F)}(p)$ which concludes this case.
3. Now if $\exists 1 \leq j \leq n \mid q_j = \Phi$ then $U(p, q_j) = \text{Coeff}_{\text{Last}(F)}(p) \otimes k$ where k is the *Null* value of some subexpression following F not depending on p .

A same reasoning can be used for the left part of the transversality. ■

Definition 16 (\mathbb{K} -balanced) *The orbit \mathcal{O} of a graph G is \mathbb{K} -balanced if G is \mathbb{K} -stable and \mathbb{K} -transverse and if there exists an input vector T of \mathcal{O} and an output vector T' of \mathcal{O} such that the orbit matrix $M_{\mathcal{O}} = T'T$. The graph G is \mathbb{K} -balanced if every maximal orbit of G is \mathbb{K} -balanced.*

Proposition 17 *A Glushkov \mathbb{K} -graph obtained from a \mathbb{K} -expression E in SNF is \mathbb{K} -balanced.*

Proof Lemma 13 enlightens on the fact that V , the tail orbit vector of \mathcal{O} , is such that $V(i, 1) = \text{Coeff}_{\text{Last}(F)}(i)$ for all $i \in P(\text{Last}(F))$, which is, from Lemma 15, the output vector of \mathcal{O} . The details of the proofs for these lemmas show in the same way that there exists an head-orbit vector and an input vector for \mathcal{O} which are equal. ■

We can now define the recursive version of WFA \mathbb{K} -balanced property.

Definition 18 *A \mathbb{K} -graph is strongly \mathbb{K} -balanced if (1) it has no orbit or (2) it is \mathbb{K} -balanced and if after deleting all edges $\text{Out}(\mathcal{O}) \times \text{In}(\mathcal{O})$ of each maximal orbit \mathcal{O} , it is strongly \mathbb{K} -balanced.*

Proposition 19 *A Glushkov \mathbb{K} -graph obtained from a \mathbb{K} -expression E in SNF is strongly \mathbb{K} -balanced.*

Proof Let G be the Glushkov of a \mathbb{K} -expression E and \mathcal{O} be a maximal orbit of G . The Glushkov \mathbb{K} -graph G is strongly stable and strongly transverse. As E is in SNF, edges of $\text{Out}(\mathcal{O}) \times \text{In}(\mathcal{O})$ that are deleted are backward edges of a unique closure subexpression F^* or F^+ . Consequently, the recursive process of edges removal deduced from the definition of strong \mathbb{K} -stability produces only maximal orbits which are \mathbb{K} -balanced. The orbit \mathcal{O} is therefore strongly \mathbb{K} -balanced. ■

Theorem 20 *Let $G = (X, U)$. G is a Glushkov \mathbb{K} -graph of a \mathbb{K} -expression E in SNF if and only if*

- *G is strongly \mathbb{K} -balanced.*
- *The graph without orbit of G is \mathbb{K} -reducible.*

Proof Let $G = (X, U)$ be a Glushkov \mathbb{K} -graph. From Proposition 19, G is strongly \mathbb{K} -balanced. The graph without orbit of G is \mathbb{K} -reducible (Proposition 10) For the converse part of the theorem, if G has no orbit and G is \mathbb{K} -reducible, by Proposition 10 the result holds immediatly. Let \mathcal{O} be a maximal orbit of G . As it is strongly \mathbb{K} -balanced, we can write $M_{\mathcal{O}} = VW$ the orbit matrix of \mathcal{O} , there exists an output vector T' equal to the tail-orbit vector V and an input vector T equal to the head-orbit vector W . If the graph without orbit of \mathcal{O} corresponds to a \mathbb{K} -expression F then \mathcal{O} corresponds to the \mathbb{K} -expression F^+ where $\text{Coeff}_{\text{First}(F^+)}(i) = W(1, i), \forall i \in P(\text{First}(F^+))$, $\text{Coeff}_{\text{Last}(F^+)}(j) = V(j, 1), \forall j \in P(\text{Last}(F^+))$. We have also $\text{Coeff}_{\text{Follow}(F^+, j)}(i) = \text{Coeff}_{\text{Follow}(F, j) \uplus \text{Coeff}_{\text{Last}(F)} \cdot \text{First}(F)}(i)$, $\forall j \in P(\text{Last}(F))$ and $\forall i \in P(\text{First}(F))$. Hence the Glushkov functions are well defined.

We now have to show that the graph without orbit of \mathcal{O} can be reduced to a single vertex. By the successive applications of the \mathbb{K} -rules, the vertices of the graph without orbit of \mathcal{O} can be reduced to a single state (giving a \mathbb{K} -rational expression for \mathcal{O}). Indeed, as \mathcal{O} is transverse, no \mathbb{K} -rule concerning one vertex of \mathcal{O} and one vertex out of \mathcal{O} can be applied. ■

5 Algorithm for orbit reduction

In this section, we present a recursive algorithm that computes a \mathbb{K} -expression from a Glushkov \mathbb{K} -graph. We then give an example which illustrate this method.

Algorithms

```

ORBITREDUCTION( $G$ )
  ▷ Input: A  $\mathbb{K}$ -graph  $G = (X, U)$ 
  ▷ Output: A newly computed graph without orbit
  1 Begin
  2   for each maximal orbit  $\mathcal{O} = (X_{\mathcal{O}}, U_{\mathcal{O}})$  of  $G$  do
  3     if BACKEDGESREMOVAL( $\mathcal{O}, T, T', Z, Z'$ ) then
  4       if ORBITREDUCTION( $\mathcal{O}$ ) then
  5         if EXPRESSION( $E_{\mathcal{O}}, \mathcal{O}, T, T'$ ) then
  6           REPLACESTATES( $G, \mathcal{O}, E_{\mathcal{O}}, Z, Z'$ )
  7         else return False
  8       else return False
  9     else return False
  10  return True
  11 End

```

The BACKEDGESREMOVAL function on \mathcal{O} deletes edges from $Out(\mathcal{O})$ to $In(\mathcal{O})$, returns true if vectors T, T', Z, Z' (as defined in definition 14) can be computed, false otherwise.

The EXPRESSION function returns true, computes the \mathbb{K} -expression E of $G' = (X_{\mathcal{O}} \cup \{s_I, \Phi\}, U')$ where $U' \leftarrow U_{\mathcal{O}} \cup \{(s_I, T(1, j), e_j) \mid e_j \in In(\mathcal{O})\} \cup \{(s_i, T'(i, 1), \Phi) \mid s_i \in Out(\mathcal{O})\}$ and outputs $E_{\mathcal{O}} \leftarrow E^+$ if \mathcal{O} is \mathbb{K} -reducible. It returns false otherwise.

The REPLACESTATES function replaces \mathcal{O} by one state x labeled $E_{\mathcal{O}}$ and connected to \mathcal{O}^- and \mathcal{O}^+ with the sets of coefficients of Z and Z' . Formally $G = (X \setminus X_{\mathcal{O}} \cup \{x\}, U)$ with $U \leftarrow U \setminus \{(u, k, v) \mid u, v \in \mathcal{O}\} \cup \{(p_j, Z(j, 1), x) \mid p_j \in \mathcal{O}^-\} \cup \{(x, Z'(1, i), q_i) \mid q_i \in \mathcal{O}^+\}$.

```

BACKEDGESREMOVAL( $\mathcal{O}, M_e, M_s, T, T', Z, Z'$ )
  ▷ Input: a  $\mathbb{K}$ -graph  $\mathcal{O} = (X_{\mathcal{O}}, U_{\mathcal{O}})$ ,  $M_e \in \mathbb{K}^{|\mathcal{O}^-| \times |In(\mathcal{O})|}$ 
  ▷ Input:  $M_s \in \mathbb{K}^{|Out(\mathcal{O})| \times |\mathcal{O}^+|}$ 
  ▷ Output:  $T \in \mathbb{K}^{1 \times |In(\mathcal{O})|}$ ,  $T' \in \mathbb{K}^{|Out(\mathcal{O})| \times 1}$ ,  $Z \in \mathbb{K}^{|\mathcal{O}^-| \times 1}$ ,  $Z' \in \mathbb{K}^{1 \times |\mathcal{O}^+|}$ 
1 Begin
2   for each line  $l$  of  $M_e$  do
3      $\text{gcd}_l(l) \leftarrow$  LEFT GCD of all values of the line  $l$ 
4     ▷  $\text{gcd}_l$  is the vector of  $\text{gcd}_l(l)$  values
5     Find a vector  $\overline{\text{gcd}_l}$  such that  $M_e = \text{gcd}_l \otimes \overline{\text{gcd}_l}$ 
6     if  $\overline{\text{gcd}_l}$  does not exist then
7       return False
8     for each column  $c$  of  $M_s$  do
9        $\text{gcd}_r(c) \leftarrow$  RIGHT GCD of all values of the column  $c$ 
10      ▷  $\text{gcd}_r$  is the vector of  $\text{gcd}_r(c)$  values
11      Find a vector  $\overline{\text{gcd}_r}$  such that  $M_s = \overline{\text{gcd}_r} \otimes \text{gcd}_r$ 
12      if  $\overline{\text{gcd}_r}$  does not exist then
13        return False
14      Find  $k$  such that  $M_{\mathcal{O}} = \overline{\text{gcd}_r} \otimes k \otimes \overline{\text{gcd}_l}$ 
15      ▷  $M_{\mathcal{O}} \in \mathbb{K}^{|Out(\mathcal{O})| \times |In(\mathcal{O})|}$  is the orbit matrix of  $\mathcal{O}$ 
16      if  $k$  does not exist then
17        return False
18       $A \leftarrow$  RIGHT GCD of all values of the  $\text{gcd}_l$  vector
19       $B \leftarrow$  LEFT GCD of all values of the  $\text{gcd}_r$  vector
20       $k_1 \leftarrow$  LEFT GCD( $B, k$ )
21      Find  $k_2$  such that  $k = k_1 \otimes k_2$ 
22      if RIGHT GCD( $k_2, A$ )  $\neq k_2$  then
23        return False
24       $T \leftarrow k_2 \otimes \overline{\text{gcd}_l}$ 
25       $T' \leftarrow \overline{\text{gcd}_r} \otimes k_1$ 
26      Find  $Z$  such that  $\text{gcd}_l = Z \otimes k_2$ 
27      Find  $Z'$  such that  $\text{gcd}_r = k_1 \otimes Z'$ 
28      delete any edge from  $Out(\mathcal{O})$  to  $In(\mathcal{O})$ 
29      return True
30 End

```

Illustrated example

We illustrate Glushkov WFAs characteristics developed in this paper with a reduction example in the $(\mathbb{N} \cup \{+\infty\}, \min, +)$ semiring. This example deals with the reduction of an orbit and its connection to the outside. We first reduce the orbit to one state and replace the orbit by this state in the original graph. This new state is then linked to the predecessors (respectively successors) of the orbit with vector Z (respectively Z') as label of edges.

Let G be the \mathbb{K} -subgraph of Figure 6 and let \mathcal{O} be the only maximal orbit of G such that $X_{\mathcal{O}} = \{a_1, b_2, c_3, a_4, b_5, b_6, c_7\}$.

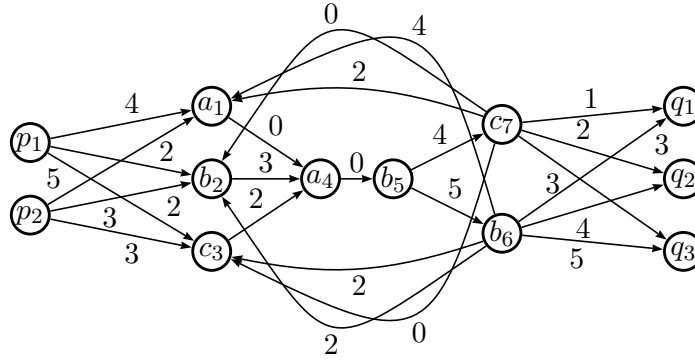
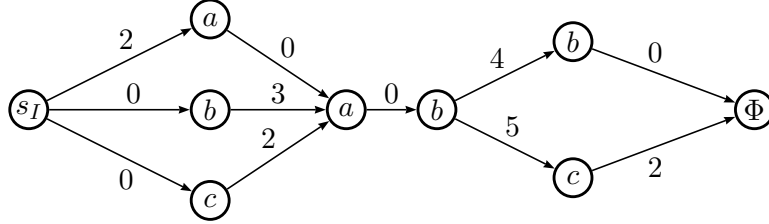


Figure 6: An example for orbit reduction

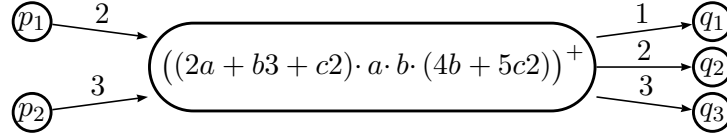
We have $M_s = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \end{pmatrix}$, $M_e = \begin{pmatrix} 4 & 2 & 2 \\ 5 & 3 & 3 \end{pmatrix}$. We can check that \mathcal{O} is \mathbb{K} -transverse.
 $M_s = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} = T'Z'$ and $M_e = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \end{pmatrix} = ZT$.

We then verify that the orbit \mathcal{O} is \mathbb{K} -stable. $M_{\mathcal{O}} = \begin{pmatrix} 2 & 0 & 0 \\ 4 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \end{pmatrix} = VW$. We easily check that the orbit is \mathbb{K} -balanced. There is an input vector T which is equal to W and an output vector T' which is equal to V .

Then, we delete back edges and add s_I and Φ vertices for the orbit \mathcal{O} . The s_I vertex is connected to $In(\mathcal{O})$. Labels of edges are values of the T vector. Every vertex of $Out(\mathcal{O})$ is connected to Φ . Labels of edges are values of the T' vector. The following graph is then reduced to one state by iterated applications of \mathbb{K} -rules.



The expression F associated to this graph is replaced by F^+ and states of \mathcal{O}^- (respectively \mathcal{O}^+) are connected to the newly computed state choosing Z as vector of coefficients (respectively Z').



6 Conclusion

While trying to characterize Glushkov \mathbb{K} -graph, we have pointed out an error in the paper by Caron and Ziadi [5] that we have corrected. This patching allowed us to extend characterization to \mathbb{K} -graph restricting \mathbb{K} to factorial semirings or fields. For fields, conditions of applications of \mathbb{K} -rules are sufficient to have an algorithm.

For the case of strict semirings, this limitation allowed us to work with GCD and then to give algorithms of computation of \mathbb{K} -expressions from Glushkov \mathbb{K} -graphs.

This characterization is divided into two main parts. The first one is the reduction of an acyclic Glushkov \mathbb{K} -graph into one single vertex labeled with the whole \mathbb{K} -expression. We can be sure that this algorithm ends without doing a depth first search according to confluence of \mathbb{K} -rules. The second one is lying on orbit properties. These criterions allow us to give an algorithm computing a single vertex from each orbit.

In case the expression is not in SNF or the semiring is not zero-divisor free, some edges are computed in several times (coefficients are \oplus -added) which implies that some edges may be deleted. Then this characterization does not hold. A question then arises: the factorial condition is a sufficient condition to have an algorithm. Is it also a necessary condition ?

References

- [1] J. Berstel and C. Reutenauer. *Rational series and their languages*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1988.

- [2] A. Brüggemann-Klein. Regular expressions into finite automata. *Theoret. Comput. Sci.*, 120(2):197–213, 1993.
- [3] A. Buchsbaum, R. Giancarlo, and J. Westbrook. On the determinization of weighted finite automata. *SIAM J. Comput.*, 30(5):1502–1531, 2000.
- [4] P. Caron and M. Flouret. Glushkov construction for series: the non commutative case. *Internat. J. Comput. Math.*, 80(4):457–472, 2003.
- [5] P. Caron and D. Ziadi. Characterization of Glushkov automata. *Theoret. Comput. Sci.*, 233(1–2):75–90, 2000.
- [6] S. Eilenberg. *Automata, languages and machines*, volume A. Academic Press, New York, 1974.
- [7] V. M. Glushkov. On a synthesis algorithm for abstract automata. *Ukr. Matem. Zhurnal*, 12(2):147–156, 1960. In Russian.
- [8] V. M. Glushkov. The abstract theory of automata. *Russian Mathematical Surveys*, 16:1–53, 1961.
- [9] U. Hebisch and H.J. Weinert. Semirings and semifields. In M. Hazewinkel, editor, *Handbook of Algebra*, volume 1, chapter 1F, pages 425–462. North-Holland, Amsterdam, 1996.
- [10] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.
- [11] S. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, Ann. Math. Studies 34:3–41, 1956. Princeton U. Press.
- [12] S. Lombardy and J. Sakarovitch. Derivatives of rational expressions with multiplicity. *Theor. Comput. Sci.*, 332(1-3):141–177, 2005.
- [13] R. F. McNaughton and H. Yamada. Regular expressions and state graphs for automata. *IEEE Transactions on Electronic Computers*, 9:39–57, March 1960.
- [14] J. Sakarovitch. *Éléments de théorie des automates*. Vuibert, Paris, 2003.
- [15] M. P. Schützenberger. On the definition of a family of automata. *Inform. and Control*, 4:245–270, 1961.